Amendments to the Claims

This Listing of Claims will replace all prior versions, and listings, of claims in this application:

Listing of Claims:

1. (Currently Amended) A tool for processing a p-code file, comprising:

analyzing p-code methods to be compiled within said p-code file;

identifying one or more p-code methods that have at least one profile parameter ~~including an associated priority level~~ above a threshold level; and

annotating said identified p-code methods to be compiled, said annotating associating a respective priority level hint with each p-code method to be compiled, ~~in a prioritized order to enable~~ said priority level hints enabling preferential processing of said p-code ~~file~~ methods in a hierarchical manner corresponding to a hierarchical order of said priority level hints ~~based on said associated priority level of each identified p-code method~~.

2. Cancelled.

3. (Previously Presented) The tool of claim 1, wherein:

said p-code file comprises an application for processing by a virtual machine (VM) just-in-time (JIT) compiler.

4. (Currently Amended) The tool of claim 1, wherein said ~~annotations~~ priority level hints are provided in-line with said identified p-code methods.

5. (Currently Amended) The tool of claim 1, wherein said ~~annotations~~ priority level hints are provided as a separate file.

6. (Original) The tool of claim 1, wherein:

said at least one profile parameter comprises at least one of a method execution time, a frequency of method invocation, a number of instructions and a use of loop structures.

7. (Original) The tool of claim 1, wherein:

said at least one profile parameter comprises at least one of an execution time parameter, an input/output utilization parameter and a processor utilization parameter.

8. (Original) The tool of claim 1, wherein: said analyzing comprises identifying at least one of a static profile parameter and a dynamic profile parameter.

9. (Currently Amended) The tool of claim 1, wherein:

said ~~annotation~~ annotating comprises setting a normally unused bit within a method access flag field of an identified Java class file.

10. (Currently Amended) The tool of claim 1, wherein:

said ~~annotation~~ annotating comprises selectively setting each of a plurality of normally unused bits within a method access flag field of an identified class file, wherein said unused bits are selectively set to define thereby ~~a~~ said priority level hint of a

respective annotated method.

11. (Currently Amended) The tool of claim 3, wherein:

each identified byte-code portion of said application is associated with one of a plurality of priority levels, said ~~annotation~~ priority level hints being indicative of respective priority levels.

12. (Previously Presented) The tool of claim 3, further comprising:

selectively pre-compiling at least a portion of said application file.

13. (Original) The tool of claim 12, wherein: said precompiled portion of said application file is included within a virtual machine.

14. Cancelled.

15. Cancelled.

16. (Currently Amended) A method of adapting the interpretation of ~~a p-code method within~~ a p-code file by a virtual machine (VM), the method comprising:

identifying one or more p-code methods within said p-code file that are annotated with a respective priority indicative annotation ~~that have at least one profile parameter including an associated priority level above a threshold level~~;

compiling p-code methods within a said p-code file ~~in a prioritized order associated~~

~~with compilation priority indicative annotation~~; and

storing said compiled p-code methods in a cache for subsequent execution in place of corresponding interpreted p-code methods, said compiled p-code methods being preferentially retained in said cache in a hierarchical manner corresponding to a hierarchical order of their respective priority indicative annotations.

17. Cancelled.

18. (Previously Presented) The method of claim 16, wherein:

said p-code file comprises an application file for processing by a virtual machine (VM) just-in-time (JIT) compiler.

19. (Currently Amended) The method of claim 16, wherein said priority indicative annotations are provided in-line with said identified p-code methods.

20. (Currently Amended) The method of claim 16, wherein said priority indicative annotations are provided as a separate file.

21. (Original) The method of claim 16, further comprising:

in response to cache memory utilization above a threshold level, prioritizing the contents of said cache memory.

22. (Original) The method of claim 21, wherein:

said cache memory contents are prioritized by deleting from said cache compiled code associated with a least recently executed method.

23. (Currently Amended) The method of claim 21, wherein:

said cache memory contents are prioritized by deleting from said cache compiled code associated with a previously compiled method having a lower priority ~~level~~ indicative annotation than a presently compiled method.

24. (Original) The method of claim 20, wherein:

compiled byte-code stored in said cache is accessed via a cache map, said cache map being updated in response to a change in cache utilization.

25. (Previously presented) The method of claim 18, further comprising:

compiling non-annotated byte-code within said application if said non-annotated byte-code utilizes VM resources beyond a threshold level.

26. (Original) The method of claim 25, wherein:

said compiled non-annotated byte-code is assigned a priority level in accordance with said utilized VM resources.

27. (Original) The method of claim 26, wherein:

said priority level of said annotated byte-code is further adapted in accordance with said utilized VM resources.

28. (Original) The method of claim 20, further comprising:

said compiled annotated byte-code is assigned a priority level in accordance with said utilized VM resources.

29. (Original) The method of claim 28, wherein:

said priority level of said annotated byte-code is further adapted in accordance with said utilized VM resources.

30. (Original) The method of claim 26, wherein said VM resources comprise at least one of an execution time parameter, an input/output utilization parameter and a processor utilization parameter.

31. (Cancelled).

32. (Cancelled).